NICE project (IST-2001-35293)

Natural Interactive Communication for Edutainment

NICE Deliverable D7.2b

Evaluation of the first NICE fairy-tale game prototype

20 April 2004

Authors

Johan Boye, Joakim Gustafson and Mats Wirén

TeliaSonera AB

Project ref. no.	IST-2001-35293
Project acronym	NICE
Deliverable status	
Contractual date of delivery	
Actual date of delivery	20 April 2004
Deliverable number	D7.2b
Deliverable title	Evaluation of the first NICE fairy-tale game prototype
Nature	Report
Status & version	
Number of pages	
WP contributing to the deliverable	WP7
WP / Task responsible	NISLab
Editor	
Author(s)	Johan Boye, Joakim Gustafson and Mats Wirén
EC Project Officer	Mats Ljungqvist
Keywords	
Abstract (for dissemination)	This deliverable describes the evaluation of the the first prototype of the NICE fairy-tale system. The evaluation is based on a data collection involving ten children who have been using the system, and involves both qualitative and quantitative measures.

Table of Contents

1 Introduction

1.1 Scope of this report

This report presents an evaluation of the first prototype of the fairy-tale system in the NICE project. Ten children have used the system, upon which they have been interviewed and asked to fill in a questionnaire. All their input has been recorded and transcribed, and all internal communication between the different modules of the system has been logged. The experimental data thus collected constitutes the basis for this evaluation.

The evaluation at this point is mainly qualitative, but we have also included some quantitative measures. First of all, we have sought to investigate to what extent children in the targeted user group (9–18 years) find the idea of a fairy-tale game fun and interesting, and which aspects of the game they like and dislike. A second goal has been to evaluate the performance of the various modules of the system, as well as the overall performance of the system. The purpose, in both cases, is to obtain information on how to proceed when designing the second prototype, rather than how to perfect the first prototype. This is because we view the first prototype not as a goal in itself, but rather as a step towards building the second (and final) prototype. Section 2 further explains the intended relationship between the two prototypes.

1.2 Structure

The report is structured as follows: Section 2 contains a description of the game scenario in the first prototype, and how it is supposed to relate to the scenario of the whole game (to be implemented in prototype 2). Section 3 describes the set-up, the workings of the system, and the experimental procedure. Section 4 is the evaluation proper.

2 Scenario

The scenario of Prototype 1 of the fairy-tale system involves a player communicating with an embodied fairy-tale character via spoken dialogue and a mouse-compatible input device to jointly solve a “put that there” problem. A comprehensive description of this scenario was given in NICE Deliverable D1.2 b, Section 2. For convenience, and because a few things have changed since the last deliverable, this section provides an accordingly updated description of the scenario.

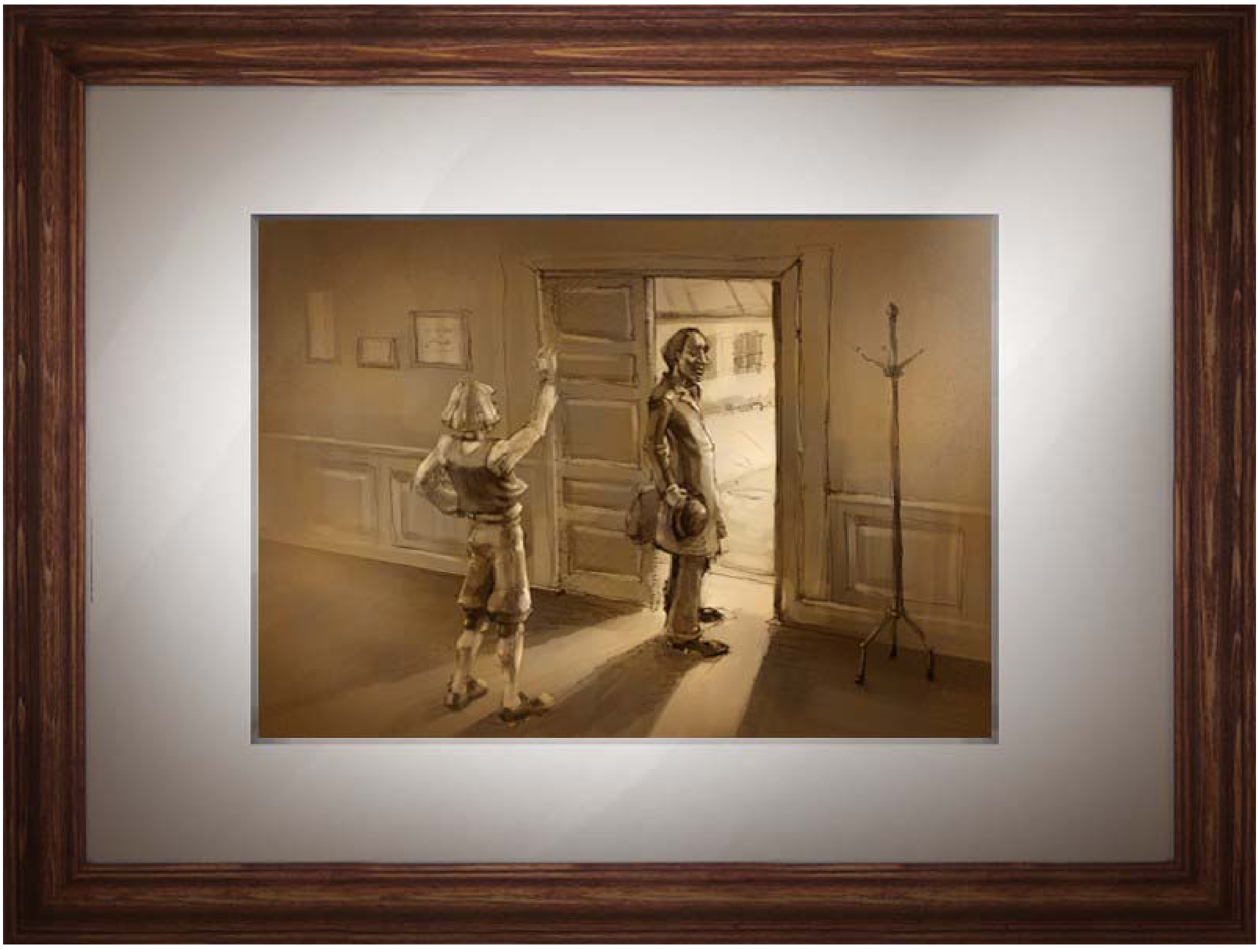


Figure 1. Initial scenario of the game: Cloddy Hans saying farewell to H. C. Andersen on his embarking to Copenhagen

2.1 Overview

The integrated NICE Prototype 1 is based on an initial game scenario which includes a single embodied character, namely, Cloddy Hans (loosely inspired by the character from H. C. Andersen’s story with the same name). Cloddy Hans is adapted as follows: He is a bit retarded, or so it seems. He cannot read and only understands spoken utterances and graphical gestures at a simple level. He does not take a lot of initiatives, but is honest and anxious to try to help the user. In spite of his limited intellectual capabilities, he may sometimes provide important clues through sudden flashes of insight. Most importantly, he is the user’s faithful assistant who will follow him/her throughout the game. Cloddy Hans has the following personality traits:

- practical rather than intellectual;
- friendly and honest;
- slow in both mind and action;
- unselfish, no goals of his own.

The character traits and inner state of Cloddy Hans are conveyed by his graphical appearance, voice and ways of expressing himself, as well as his actions, gestures, facial expressions and postures. The intent is to give the player the impression of a believable fairy-tale character with a distinct personality, and notably one that it is challenging for the player to get to know independently of the problem-solving dimension of the game.

The game begins in H. C. Andersen's house in Copenhagen, Denmark in the 19th century. Andersen has just left on a trip to Odense, and has asked one of his fairy-tale characters, Cloddy Hans, to guard his fairy-tale laboratory while he is away (see Figure 1). The key device in the laboratory is a fairy-tale machine, which nobody except Andersen himself is allowed to touch. On a set of shelves beside the machine, various objects, such as a key, a hammer, a diamond and a magic wand, are located (Figure 2). When a set of objects is taken from the shelves, put into suitable slots in the machine and a lever is pulled, the machine will construct a new fairy-tale in which the objects come to life.

However, just before the user enters the game, Cloddy Hans has violated the rules by taking one of the objects and putting it into the machine. As nothing harmful has happened, Cloddy Hans gets the idea of surprising H. C. Andersen with a new fairy-tale on his coming back. There is a problem, however: Each slot is labelled with a symbol which tells which type of object is supposed to go there, but since Cloddy Hans is not very bright, he needs help from the user with understanding these. There are four slots, which are labelled with symbols denoting "useful", "magical", "precious" and "dangerous" things, respectively. Which object goes in which slot is sometimes more obvious (provided you understand the symbols), like the diamond belonging in "precious", and sometimes less obvious, like the knife belonging in "useful" rather than "dangerous".

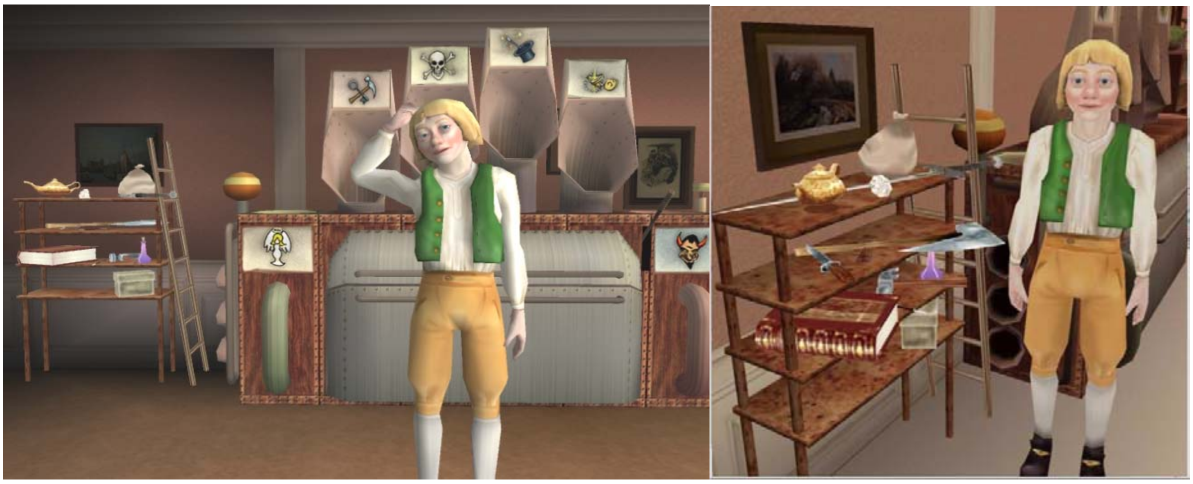


Figure 2. Inside the game: Cloddy Hans by the fairy-tale machine and the shelf with objects.

Although the initial scenario is the only one present in Prototype 1, it plays an important role and serves as the entry to the game also in Prototype 2. Basically, the scenario in Prototype 1 can be seen as a subdomain of the subsequent scenario in Prototype 2. The latter takes place in the fairy-tale world proper, but includes objects as well as Cloddy Hans himself from the initial scenario (for more description, see below).

2.2 Role of the user

The user perceives the fairy-tale world through a first-person perspective. Hence, there is no user avatar, but the user is still perceived as appearing in the world by other characters in the game. The user's means of action in the world are:

- Speaking to other characters in the game (in Prototype 1, only Cloddy Hans).
- Pointing and gesturing at arbitrary objects and locations.

The available output modalities are as follows in the two prototypes:

- Speech and/or gesture, facial expression, gaze, posture.

Note that contrary to what was forecasted in Deliverable D1.2 b, facial expression, gaze and posture have been introduced already in Prototype 1. The same also goes for synchronized lip movements.

Various 3D objects appear in the environment, most importantly the shelves with fairy-tale objects beside the machine. The user can ask Cloddy Hans to manipulate objects by referring to them verbally and/or by using the mouse, but cannot manipulate them himself.

To understand the reason for these rather limited capabilities of the user (in particular, the lack of direct manipulation), we have to recall what distinguishes NICE from other games, namely, spoken multimodal dialogue. We thus want to ensure that multimodal dialogue is appreciated by the user not just as an "add-on" but as *the primary means of progressing in the game*. Our key to achieving this is to deliberately limit the capabilities of the key actors — the user and Cloddy Hans — in such a way that they can succeed only by cooperating through spoken multimodal dialogue. In other words, the user is intelligent but cannot himself affect objects in the world; Cloddy Hans on the other hand is a bit retarded but capable of physical action according to what he gets told (and he may occasionally also provide tips to the user).

2.3 Purposes of the initial scenario

The initial scenario has been carefully designed to serve the following purposes:

1. It is a "grounding game" set in the context of a very limited "put that there" task. Thus, its real purpose is to let the user and Cloddy Hans agree on what the 3D fairy-tale objects can be used for and how they can be referred to. This process also lets the player find out (by trial-and-error) how to adapt in order to make it easier for the system to understand him or her. For example, one conclusion that the user might reach is that it is more efficient to use multimodal input to Cloddy Hans instead of just spoken utterances.

2. We can let the subsequent game in the fairy-tale world depend on what objects have been chosen by the user in the initial scenario. The advantage of this is that the objects

are already “grounded”; for example, a sack of gold will be visually recognized by the player and there is an already agreed way of referring to it.

3. Since the initial scenario (limited to a simple task as well as conventionalized social greetings) is a subdomain of the domain in Prototype 2, it allows us to collect data that can be used for doing real-time user adaptation in the fairy-tale world. For example, if the user consistently makes use of a single modality, Cloddy Hans may ask a clarification question in such a way that the other modality is likely to be elicited. Thus, if the utterances do not include gestures, Cloddy Hans might ask the user to literally point at the relevant object for clarification. It is also not certain that Cloddy Hans understands utterances like ”left” and ”right”.

2.4 Subsequent game scenario

In the integrated NICE Prototype 2, the initial scenario described above will be augmented with a subsequent scenario, taking place in the fairy-tale world proper. A major difference between the initial and augmented scenarios is that the latter will involve multiple tasks, multiple locations and several additional fairy-tale characters. Figure 3 shows two of these characters, namely, the Prince and Thumbelina, together with Cloddy Hans. Figure 4 shows the key locations (the village, the bridge and the wind mill).

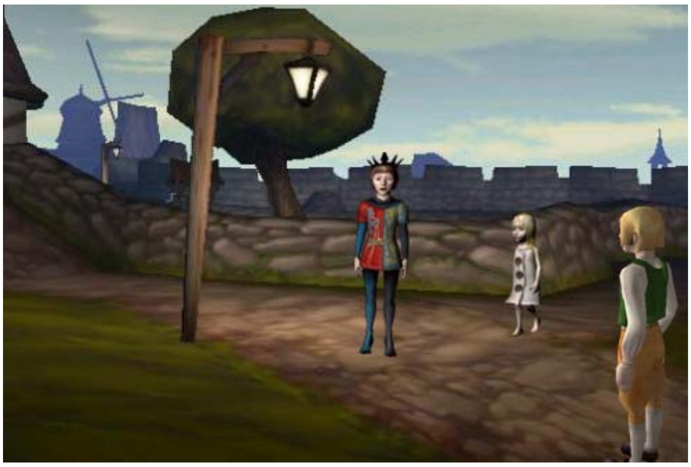


Figure 3. The fairy-tale world (just outside the village) and some of the characters.

The basic idea for the augmented game scenario is to let the player obtain a piece of information from a key figure, such as the Prince, which requires the player to go to a different location to perform an action (such as bringing back a valuable object or obtaining a piece of information). Throughout, Cloddy Hans will act as the player's assistant whose task it is to help the player. For example, the starting scene of the subsequent scenario might take place at the wind mill (see Figure 4). Something happening there might suggest to the player that he or she should go to the village. However, to do so she must pass the bridge, which is guarded by a witch. Each time the player wishes to cross the bridge, she must solve a problem presented by the witch in order to be let through. As mentioned above, various objects from the initial scenario will re-occur and will be key to solving the problems in this scenario.

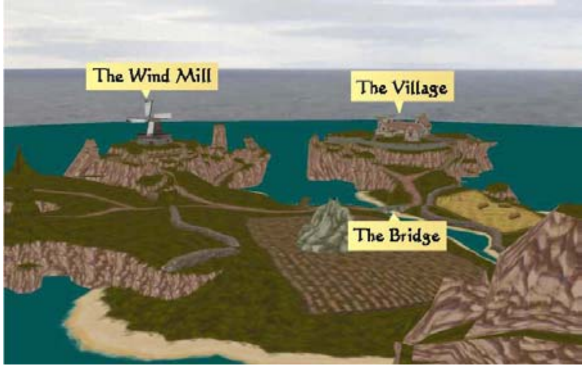


Figure 4. An overview of the fairy-tale world.

3 Data collection

3.1 Set-up

The data collection was carried out at the Technical Museum in Stockholm. The system was displayed on a large back-projection screen (see Figure 5). The user could give input to the system by means of a wireless microphone headset and a wireless gyro mouse[1]. The system was supervised, and partly controlled, from a neighboring room (see Figure 6).

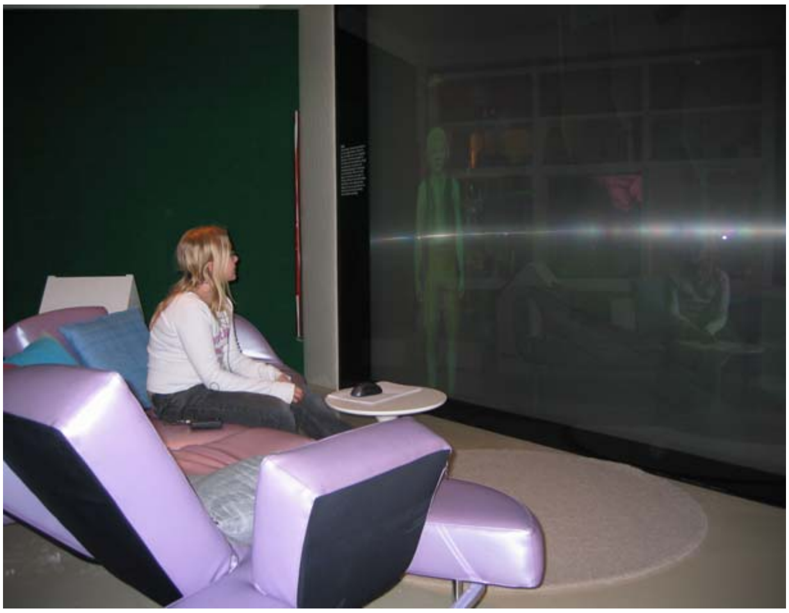


Figure 5. A young user talking to Cloddy Hans.

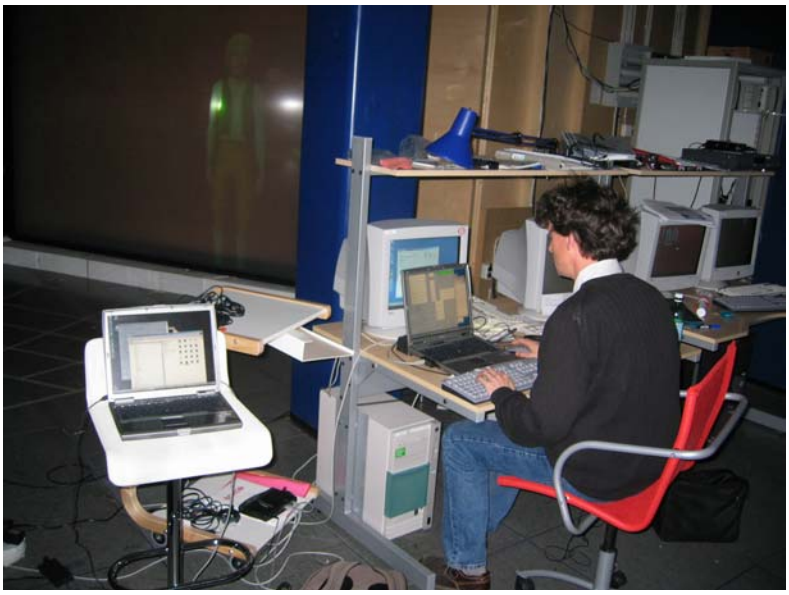
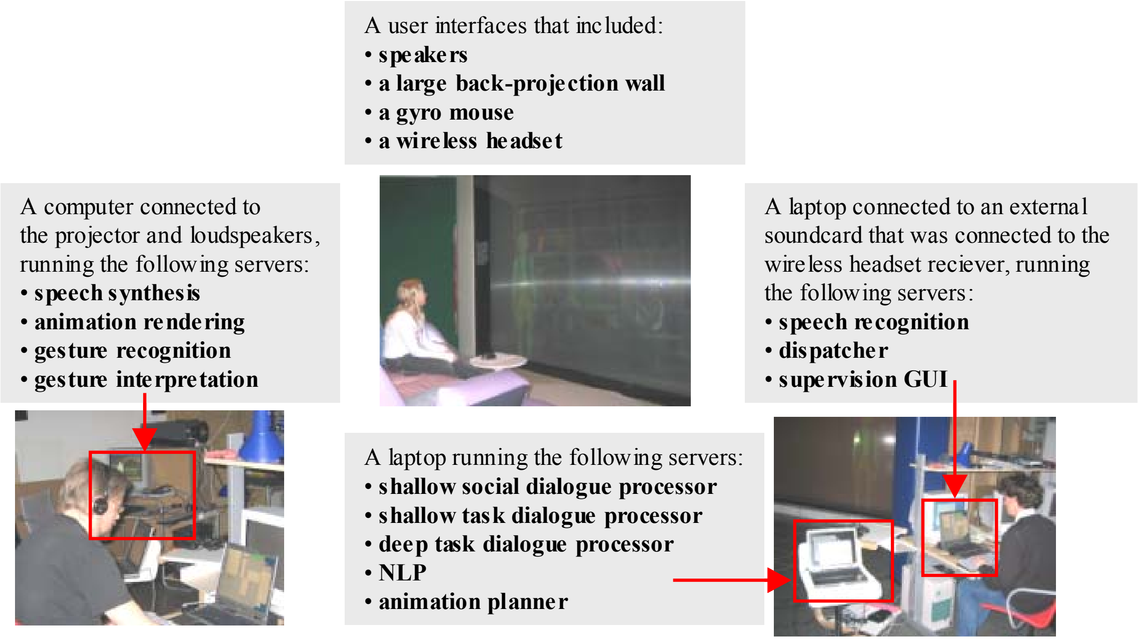


Figure 6. System supervision from behind the scene.

[1] A gyro mouse is a pointing device that can be held freely in the air. If put on a table, it can also be used as an ordinary mouse.

The system set-up is depicted in Figure 7. The very large back-projection screen made it possible to display Cloddy Hans in natural size (180 cm tall). The sound came from speakers built into the ceiling above the screen. The gyro mouse did not seem to pose any problems for the users. A professional wireless headset for usage in television shows and concerts had been chosen to ensure the sound quality. Prior to the data collection, it was tested in TeliaSonera's sound lab and was found to have excellent frequency characteristics. The radio submission part of the wireless headset did not affect the sound quality. The system was robust to both user movements and presence of electrical equipment. To increase sound quality and decrease the influence of noise from the computer's fans and transformers, an external soundcard with built-in amplifiers was used. The soundcard was connected to a laptop via a firewire cable.

The system modules were run on one stationary computer with a high-performance graphics card, and on two laptops. The reason for using more than one computer was twofold: in made the system faster, and it made it possible to supervise and intervene the systems automatically generated output. If the system would not have been run in supervised mode it could have been run only on the stationary computer without significant degradation in speed.



A user interfaces that included:
- **speakers**
- **a large back-projection wall**
- **a gyro mouse**
- **a wireless headset**

A computer connected to the projector and loudspeakers, running the following servers:
- **speech synthesis**
- **animation rendering**
- **gesture recognition**
- **gesture interpretation**

A laptop connected to an external soundcard that was connected to the wireless headset reciever, running the following servers:
- **speech recognition**
- **dispatcher**
- **supervision GUI**

A laptop running the following servers:
- **shallow social dialogue processor**
- **shallow task dialogue processor**
- **deep task dialogue processor**
- **NLP**
- **animation planner**

Figure 7. An overview of the different parts of the system.

3.2 System

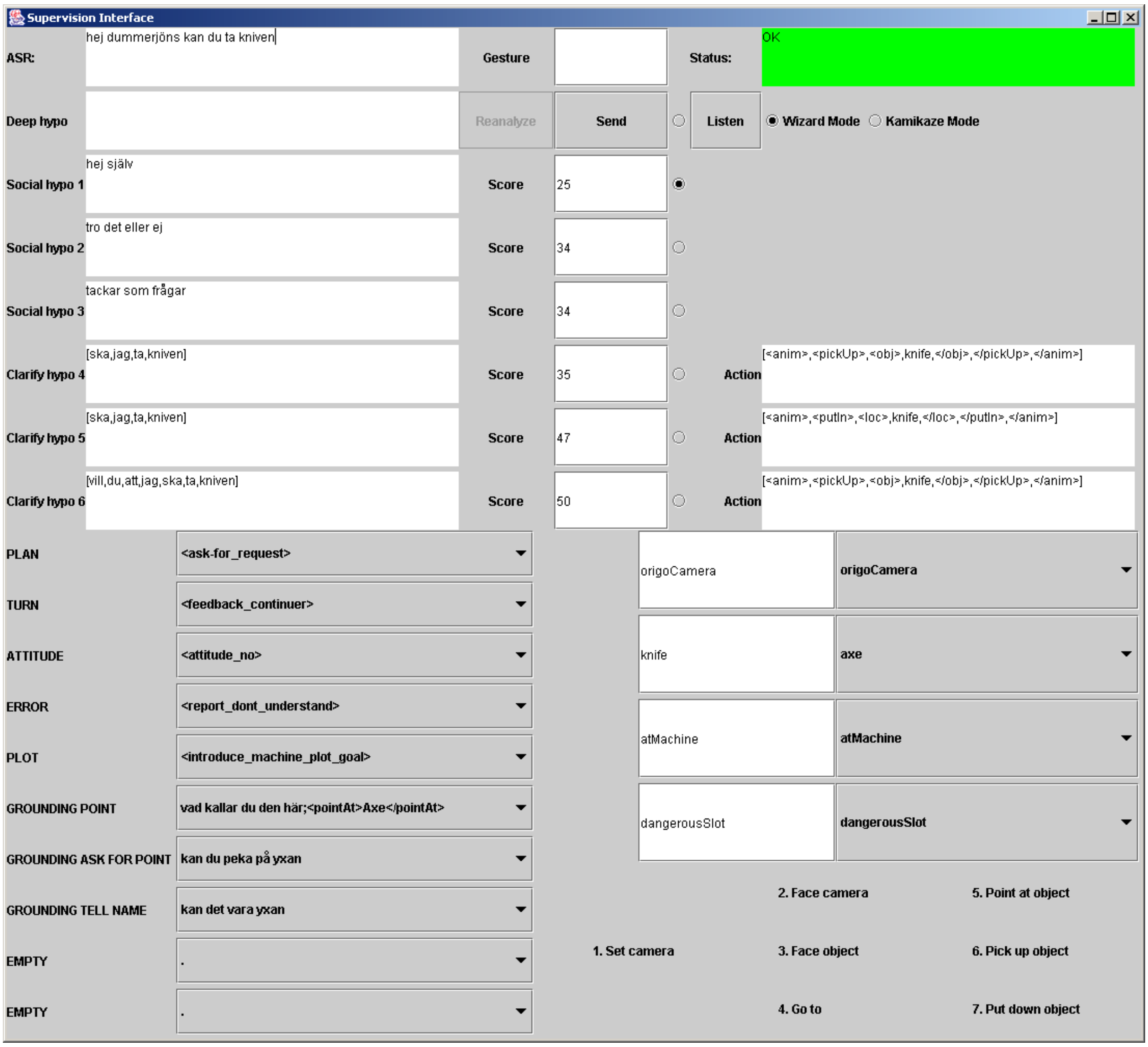
3.2.1 Supervised mode

For the purpose of data collection, the system was run in *supervised mode*. This means that in each turn, the system computed an n-best list of possible Cloddy Hans utterances, and an n-best list of possible actions (animations). The human operator then had several options:

1. Go with the system's primary suggestion by pressing a "Send" button.
2. Select some suggestion lower down the n-best list by clicking on that suggestion and pressing "Send".
3. Select predefined utterances and actions from a number of pop-up menus.
4. Edit the speech recognition result (which was displayed in a window) and let the system reanalyze. This option was particularly useful in the cases where some crucial word in the input had been misrecognized, leading the succeeding processing astray.
5. Type in an utterance to be synthesized in a free-text window.

Obviously, alternative 1 was faster than alternative 2, which in its turn was faster than alternative 3, and so forth. In order to facilitate a real-time conversation, the operator only rarely used the alternatives 4 and 5.

A supervised system, such as the one described above, thus represents a middle ground between a fully automatic system on the one hand, and a completely simulated system (full Wizard-of-Oz) on the other. In fact, for sophisticated systems like the NICE fairy-tale system, complete simulation is out of the question. To give the operator the slightest chance of holding up the system's end of the conversation, a high degree of automation is necessary.


Supervision Interface

ASR: hej dummerjöns kan du ta kniven
Gesture
Status: OK
Deep hypo
Reanalyze Send ○ Listen ● Wizard Mode ○ Kamikaze Mode

Social hypo 1	hej själv	Score	25	●		
Social hypo 2	tro det eller ej	Score	34	○		
Social hypo 3	tackar som frågar	Score	34	○		
Clarify hypo 4	[ska,jag,ta,kniven]	Score	35	○	Action	[<anim>,<pickUp>,<obj>,knife,</obj>,</pickUp>,</anim>]
Clarify hypo 5	[ska,jag,ta,kniven]	Score	47	○	Action	[<anim>,<putIn>,<loc>,knife,</loc>,</putIn>,</anim>]
Clarify hypo 6	[vill,du,att,jag,ska,ta,kniven]	Score	50	○	Action	[<anim>,<pickUp>,<obj>,knife,</obj>,</pickUp>,</anim>]

PLAN	<ask-for_request>
TURN	<feedback_continuer>
ATTITUDE	<attitude_no>
ERROR	<report_dont_understand>
PLOT	<introduce_machine_plot_goal>
GROUNDING POINT	vad kallar du den här;<pointAt>Axe</pointAt>
GROUNDING ASK FOR POINT	kan du peka på yxan
GROUNDING TELL NAME	kan det vara yxan
EMPTY	.
EMPTY	.

origoCamera	origoCamera
knife	axe
atMachine	atMachine
dangerousSlot	dangerousSlot

Figure 8. The supervision interface

3.2.2 Speech recognition

Since the user and Cloddy Hans are supposed to collaborate to solve tasks, it is important that there is a symmetric relationship between the system's input and output capabilities. Since the wording of system prompts tends to influence users, it is important to assure that the system is able to understand everything it can produce. Therefore the speech recognition is trained on the things Cloddy Hans can say, as well as on user utterances collected from human–computer interactions. The speech recognizer used a statistical bi-gram language model. The basis for this language model is a domain model, which also underlies Cloddy Hans's repertoire of utterances, including names for all objects in H. C. Andersen's study, and verbs for all the operations Cloddy Hans can perform (like picking up things, moving about, etc.). It also includes clarification questions to all task-oriented utterances. A set of general dialogue handling utterances were also designed; these include utterances for grounding, error handling, attitudal feedback and turn regulation utterances. Data from test persons using earlier versions of the systems has been added to model, as well as typical socializing utterances from corpora collected with our earlier systems, like August (Gustafson and Bell 2000) and Pixie (Bell 2003).

3.2.3 Speech synthesis

Since the system will be used for collaborative grounding dialogues, it is also important that the system is able to say everything it had been designed to understand. In the task-oriented dialogues it is thus important that Cloddy Hans can talk about the physical actions he will have to perform to solve a certain task. In the small-talk domain he must be able to ask the users the same questions that the system has been prepared to answer. Another design criterion was that Cloddy Hans should be able to understand and generate both grounding and turn regulation utterances. Finally, to facilitate rephrasing as an error-handling strategy, all domain and meta utterances have been provided in a version with alternative wording.

Cloddy Hans's verbal output includes the following types: *responses to instructions from the users* (confirmations; acknowledgements and clarification); *initiatives to fulfill plans; social utterances*, *meta utterances* (grounding and error handling), *attitudal feedback* and *turn regulation utterances* (floor-holders, back-channels and filled pauses). The turn regulation utterances, attitudal feedback and extra-linguistic sounds are used to buy time while the system generates the next Cloddy Hans utterance, but more importantly, they are used for the purpose of conveying his uncertain personality.

A number of domain utterances have been designed that let Cloddy Hans explain the overall plot as well as talk about the task at hand. To facilitate grounding, Cloddy Hans has been given the possibility to ask clarification questions about everything the user can ask him to do. The recorded corpus includes sentences where all objects and slots have been placed in both medial and final position of the utterances, making it possible to ask clefted clarification questions like *"Is it the axe you want me to put in the useful slot?"* as well as *"Was it in the useful slot you wanted me to put the axe?"*. All utterances were also tagged with prosodic phrase boundaries and emphasized words prior to recording. Utterances with slots and objects in all combinations of position and emphasis have been recorded. Finally a number of sentences with only function words were recorded to make sure that they were sufficiently covered, and to increase the coverage of Swedish di-phones a set of old Swedish sayings were recorded.

A new corpus-based unit selection synthesizer, **Snacka,** has been developed by Kåre Sjölander at CTT/KTH in close collaboration with the Voice Technologies group at TeliaSonera (Gustafson & Sjölander 2004). Its unit-selection synthesizer has been implemented as an extension to the Snack sound toolkit (Sjölander and Beskow 2000). The synthesizer only requires a set of speech recordings with matching orthographic transcriptions files. With this as input the system is able to generate a synthetic voice without further manual intervention. The aim in creating the system was to have symmetric analysis and synthesis components. Modules for tasks such as letter-to-sound, text analysis, co-articulation modelling, have been designed with both tasks in mind and are used by both components. In this way it is assured that there is a high degree of match between unit database and synthesis output which increases synthesis quality. A special quality assessment tool has been developed that makes it easy to check domain sentences. Sentences that sound strange can be examined, automatic segmentation errors can be corrected, and bad units can be pruned manually. These changes can be used instantly for speech synthesis, without rebuilding the whole voice in most cases. It is also possible to perform acoustic voice transformations in order to better match the personality of the fairy-tale characters. This is done on the whole recorded voice corpus prior to creating the voice. The implementation of the Snacka synthesis system will be described in Gustafson and Sjölander 2004. The system also automatically generates lip-synchronisation tracks for the animation system, as well as time stamping of animation tags that are inserted in the output text.

3.2.4 Animation interface

Cloddy Hans is currently able to perform the following non-verbal behaviors: *physical action* (e.g. point at object, pick up object and put object in slot); *emotional display* (e.g. surprised, angry); *state of mind* (i.e. idle, listening, thinking), *turn regulation cues* (i.e. nodding or look away from user), *back-channelling gestures* (raising eyebrows, nodding).

The animation system, implemented by Liquid Media, provides a large repertoire of small animations and actions. An instruction like "go to the machine" (which is treated as an atomic operation by the dialogue modules) is translated into a sequence of such small animations and actions: (1) Change camera angle and show Cloddy Hans from a distance; (2) Make Cloddy Hans walk up to the machine; (3) Change camera angle again; and (4) Make Cloddy Hans turn to the camera. An instruction like "pick up the sword" is translated into an even more complex sequence of animations.

The module performing the translation from instructions like "pick up", "go to" etc., into sequences of animation requests is called the Animation Planner. This module is also responsible for the direct communication with the animation system. All animation requests produced by the Animation Planner is put in a queue. A separate thread reads a request from the queue one at a time, sends the request to the animation system, awaits a "request done"-message, and the proceeds to the next message in the queue.

The lip-synchronization animation system currently uses a set of 12 visemes, and the lip-synchronization track is generated by the speech synthesizer module. All meta, grounding, turn regulating and attitude feedback utterances have also been tagged with emotional information in order to chose eyebrow poses that support emotional content of the verbal part of the utterances (sad, angry and surprised eyebrow shapes). Turn handling gestures (like nods, eye movements, eye brow raises) that increased the perceived reactivity of the system were generated automatically by the system as a result of start-of-speech events and recognition rejections from the speech recognition server.

3.2.5 Shallow dialogue processing

In order to compute the n-best list of possible responses, the system makes use of a number of shallow dialogue processing modules. Here, "shallow" means that the module in question does not perform any sophisticated reasoning involving the dialogue context (the preceding dialogue) when preparing the answer.

One such shallow processor always sought to interpret the user's input as a socializing utterance. This module was added in order to catch utterances like "Hi there", "How are you?", "You are stupid!" etc., producing answers like "Hi there too", "I'm fine, thanks", and "That wasn't a very kind thing to say". Another shallow processor tried to interpret the user's utterance as a command to do something; pick up an object, point at something, go to a specific location etc., and responded with the appropriate animation instruction. Yet another processing track sought to produce counter-questions like “Do you want to pick up the diamond?”, “Do you want me to go to the machine?”, etc.

Each shallow processor was implemented as a table, mapping input patterns (possible user utterances) to output strings (Cloddy Hans utterances and/or actions). Each output string was given a penalty score, which increased the more the corresponding input pattern differed from what the user had actually said. Thus, for each user utterance, the output strings corresponding to Cloddy Hans’s reaction could be rated according to their score. The best three suggestions (i.e. the ones with the lowest score) were then displayed in the supervision interface.

The suggestions produced by two different shallow processors could also be compared using the scoring mechanism. As it turned out, if suggestion X produced by processor A had a lower penalty score than suggestion Y produced by processor B, then more often than not suggestion X was the more appropriate (see further Section 4).

3.2.6 Deep dialogue processing

Besides the different shallow dialogue processing tracks, the input was also processed by a “deep” dialogue processing track, implemented according to the ideas outlined in deliverables D1.2b, D3.5 and D5.1b.

3.2.7 Gestural input

The Gesture Recognizer (GR) and Gesture Interpretation (GI) modules implemented by LIMSI (see NICE deliverable D3.4) were also connected to the system. The output from the GI module is the name of the object the user has pointed at (or otherwise gestured at) with the gyro mouse. This information was used for automatic focus management in the supervision interface. If, for instance, the GI reports that the user has clicked on the magic wand, and the operator the presses the “Pick up” button on the supervision interface, Cloddy Hans will pick up the magic wand.

4 Evaluation

Ten subjects in the ages of 11 to 15 years were recorded. 650 utterances were recorded and transcribed in all. Out of these, 600 were judged as being directed to the system (rather than own-communication, initial questions to the experimental supervisor, etc.). The following evaluation is based solely on this set of 600 system-directed utterances. (For a more detailed description of the user data see, D.2.2.) The subjects provided information about themselves prior to the data collection, and filled in a questionnaire afterwards. They were also interviewed after their interaction, where they were asked to give more details about what they thought about the system and their interactions with it.

4.1 User questionnaire

The users were asked to select words that they felt described Cloddy Hans well from a fixed list of adjectives (irritating, friendly, slow, fast, helpful, human, robot-like, and so on), as well as select words from another list to describe their view on the whole game (interesting, boring, too long, too short, easy, hard, and so on). The most commonly used words for Cloddy Hans's person were the following:

- Friendly
- Slow
- Helpful
- Stupid
- Cute
- Robot-like

Apart from the last one (robot-like), these are in accordance with the personality traits that we had aimed for when designing his persona (see 2.1 above). The robot-like feature probably comes from the fact that we used a very early version of the gesture system, in which Cloddy Hans can only move one limb at a time, and where consecutive animations are not blended. This made Cloddy Hans give a somewhat mechanical impression. When the animation track system is fully implemented it will be possible to move different body parts simultaneously, as well as blend between animations on the same body part. Then it will be possible to give Cloddy biological idle movements as well as reactive awareness movements along with the lip synchronization and deliberate body gestures.

Even though the users found Cloddy to be slow and a bit stupid and the task a bit boring, they enjoyed using the game, mostly because they could talk to Cloddy and that he talked back and did things that they told him to do. As already mentioned, the users were asked to choose words that they thought described the whole system. The distribution of positive and negative words are shown in Figure 9 below.

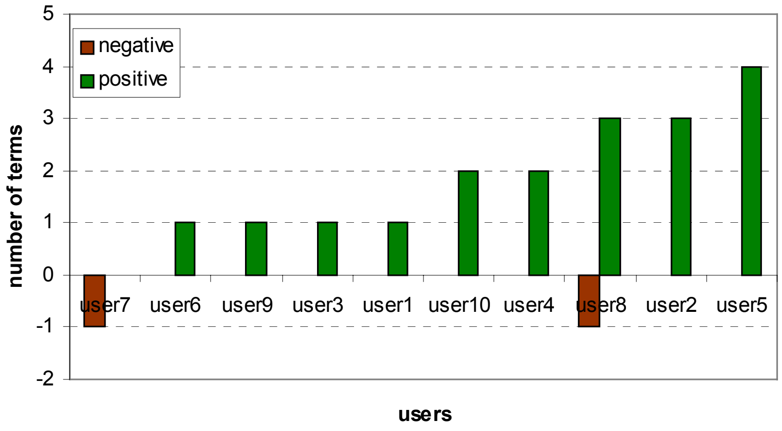


Figure 9. The number of words the users chose to describe the system as a whole

4.2 Speech recognition

The bi-gram grammar for the speech recognizer was primarily built on sentences designed from a model of the task, verbal descriptions of the objects in the visual scene and from experiences from previously developed spoken dialogue systems. This means that the first iteration suffers from holes in the coverage, both in terms of words and in terms of syntactic structures. The speech recognizer used in the supervised data collection described in this report had a word error rate (WER) of 58%, while the concept error rate was 37%. The WER was highly user-dependent ranging from 37% to 90%. There is a tendency that the older children get better results than the younger.

Subject	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Gender	**female**	**female**	**male**	**female**	**female**	**male**	**male**	**male**	**female**	**female**
Age	**14**	**14**	**15**	**14**	**14**	**15**	**14**	**12**	**11**	**11**
WER	76%	37%	46%	68%	51%	48%	37%	83%	90%	48%

Figure 10. The Word Error Rates per user.

If the recorded 600 sentences are added to the original training corpus of 1463 sentences, the word error rate decreases to 43% and the concept error rate dropped to 24%. This is not very surprising since the test files now are part of the training set. It is presented in this early system evaluation only as an indication on what an improved bi-gram grammar for the recognizer might do, and to see how well a bi-gram grammar with full coverage performs on spoken input from 11-15 year olds within our task. The improvement on a word level is shown in Figure 11.

	insertions	**deletions**	**substitutions**
old grammar	232	185	801
new grammar	185	173	540

Figure 10. The total number of errors for the two bigram grammars

173 utterances that got lower word error rates, 395 was not changed at all, and only ten got worse word error rates. These ten words did not have worse concept error rate (i.e. it was not

the information parts that were lost). The 173 utterances that got improved word recognition were analyzed further to see how the improved word recognition influenced the concept correctness. The result is shown in Figure 11.

		New grammar		
		wrong	**partly correct**	**correct**
Original Grammar	**wrong**	6.9%	2.9%	31.2%
	partly correct	0%	4.6%	24.9%
	correct	0%	0.6%	28.9%

Figure 11. The number of utterances with correct concept for the bi-gram grammars.

The table shows that it often was recognition of the information-carrying words that improved with the new grammar. The 98 utterances that got improved concept error rate were further analyzed and the improvements were categorized into five categories (See Figure 12).

new command	new object name	single word	user name	better bigrams
11	20	10	3	54

Figure 12. The improvement categories in the updated ASR grammar.

Improved bi-gram statistics made the greatest difference, for example making the recognizer correctly choosing “ja”(yes) instead of “jag”(I) 22 times more often. In 20 cases the subjects had referred to objects in ways that were missing in the original grammar, and in 11 cases the users asked Cloddy Hans to do things that had not been included in the system (i.e. “hoppa”(jump) and “gå rakt fram”(go straight ahead)). In 11 cases, new words for non-task concepts were encountered (i.e. “seg”(slow) and “bravo”(great)).

4.3 Shallow dialogue processing

The shallow dialogue processing track was evaluated on the full set of data (600 utterances) by the help of a human expert who, for each turn, determined whether the selected answer was appropriate in the given dialogue context. The results for users 1–10 are shown in Figure 13 below. The “ASR” row indicates in how many turns the shallow processing track selected the appropriate response, given the recognition result (the average result in this row is 30%). The “Transcr.” row indicates how well this module would have done given perfect recognition (here the average is 39%). The figures should not be taken too seriously, as this the first design iteration, and the shallow dialogue processor was constructed without access to real data.

Subject	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Gender	**female**	**female**	**male**	**female**	**female**	**male**	**male**	**male**	**female**	**female**
Age	**14**	**14**	**15**	**14**	**14**	**15**	**14**	**12**	**11**	**11**
ASR	20%	30%	34%	41%	29%	54%	32%	12%	10%	36%
Transcr.	33%	36%	53%	46%	44%	59%	50%	18%	18%	34%

Figure 13. Shallow dialogue processing results

4.4 Parsing

We evaluated the parser in the deep processing track on the same set of data but excluding 100 utterances that had been completely rejected by the speech recognizer. Thus, for this evaluation 500 utterances were used. The results are summarized in Figure 14.

		parse correct		
		wrong	partly corr	correct
ASR correct	wrong	16,6%	0	0
	partly corr	2,5%	17,2%	0
	correct	8,2%	13,4%	42,2%

Figure 14. Deep parsing processing results

The three rows in the table represent speech recognition results (on the concept level), whereas the columns represent parsing results. The part of the matrix above the main diagonal is zero, since the parser obviously can not detect a concept which is not present in the recognizer output. On the other hand, coverage leaks in the parser may result in the parser not picking up all the concepts mentioned in the recognizer output. So, for instance, 27,3% (16,6 + 2,5 + 8,2 %) of the parser output is completely wrong, although only 16,6% of the recognizer output is completely wrong. Again, the reader should keep in mind that the parser grammar was constructed without any access to real data.

5 References

Gustafson, J, and Bell, L (2000) "Speech Technology on Trial: Experiences from the August System", *Journal of Natural Language Engineering: Special issue on Best Practice in Spoken Dialogue Systems*.

Bell, L (2003): "Linguistic adaptations in spoken human-computer dialogues – Empirical studies of user behavior", PhD thesis KTH.

Gustafson J. and Sjölander, K. (2004) "Voice creation for conversational fairy-tale characters", submitted to 5th ISCA Speech Synthesis Workshop, Carnegie Mellon University 14-16 juni 2004.

Sjölander K and Beskow J. (2000) "WaveSurfer - an Open Source Speech Tool," In Proceedings of ICSLP, 2000.